TCP/IP Implementation Using Embedded Controller

S. Ganesh

Sri Muthukumaran Institute of Technology, Chennai-69

ABSTRACT: This paper deals with the design of an Embedded Web Server based on MCS51Family Controller. The designed embedded system hosts a small w eb page, which is served with the dynamic data upon the client http request. The entire system is developed around MCS 51 and Ethernet development board. This system enables a user to monitor and control a physical quantity from a remote location through Internet.

Keywords—*TCP/IP, Embedded, Ethernet, Simple Mail Transfer Protocol (SMTP).*

INTRODUCTION

An Embedded system is device that has computer intelligence and dedicated software to perform a single task, or group of related tasks. Embedded Systems are often used to do monitoring and controlling functions such as gathering sensor readings and controlling relays and motors.

Ethernet is the technology, used to connect computers in home and offices. It is also possible to interconnect networks by router and Gateways. The work done in this paper bridges the gap between the Ethernet technology and Embedded Systems. One development that has made Ethernet feasible for embedded systems is the availability of inexpensive controller chips to handle the details of Ethernet communications. The CPUs that provide the processing power for embedded systems have also faster and more capable of handling the demands of communicating with the controllers. Internet protocols provide standard, well-documented ways of exchanging data. Both Ethernet and the Internet protocols are free and open standards available for use without royalties or licensing fees.

EMBEDDED WEB SERVER

Normally Web Server is a computer in which lot of web pages are stored. Web Server for hosting a corporate web page should have the capability to handle more traffic.

The aim of this work is to neither hosting a corporate web pages nor handling much traffic.

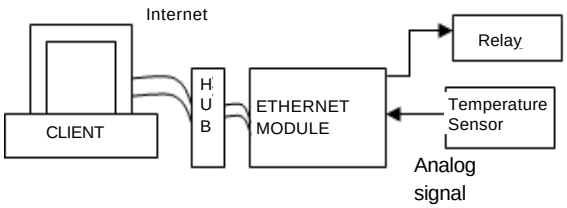

Fig. 1: Block diagram representation of Embedded Web Server

An embedded system that functions as a Web server generally has all of the following:

- Non-volatile memory to hold pages to be served.
- Support for TCP and IP. Requests for Web pages and the pages sent in response travel in the data portion of TCP segments.
- Support for HTTP. The server must be able to understand and respond to receive requests for Web pages. The HTTP standard specifies the format for the requests and replies.
- A local-network or Internet connection. To serve pages on the Internet, the Web server must have an Internet connection. Any firewalls must be configured so the system can receive HTTP requests.
- One or more pages to serve. The Web pages are files or blocks of text that use a form of encoding called hypertext markup language (HTML). The HTML encoding specifies the for matting of text and images on the page, including text size and fonts and the positioning of text and other elements on the page. The HTML code may include links to images that appear on the page, as well as links to other pages or resources. In serving a Web page with dynamic content, the software must have a way of inserting the dynamic content as the page is being served.

SOFTWARE DESIGN

The heart of the Embedded Web server is the software, which is running in it. In this work using Dynamic C™ Integrated Development Environment develops the software. To make the Controller communicate to Internet, TCP/IP stack should be implemented in the software. The software development in Dynamic C is easier, as it comes with TCP/IP stack.

Many Web pages are static. That is the content of web page remains the same for every http request. It doesn't have any dynamic content, like date or time, the value of which is changing. Embedded Web servers are often used to display the real time information such as sensor reading

from a remote location. This necessitates that the web page is served with dynamic content. Here the dynamic content is the sensor reading, which may change with time.

To deliver the web page with dynamic content, Server Side Include (SSI) directives are used. It instructs the server to insert the values of variable in the appropriate locations in the web page. The following HTML code includes SSI Directives.

```
<html>
<head>
<title>Temperature Monitor</title>
</head>
<body>
<h1> Temperature Monitoring Demo</h1>
<p>Temperature: <!--#echo var="Temperature"-->
</p>
</body>
</html>
```

A Server Side Include directive uses the same delimiters as a HTML comment. A comment, which is text that the browser ignores and doesn't display, is enclosed by <!-- and -->. On receiving a page that contains an HTML comment, the browser displays the page the same as if the comment and its delimiters weren't present.

Another use for comment delimiters is to enable a page to specify Server Side Include (SSI) directives that the server executes before serving the page to the browser. Before serving a page containing an SSI directive, the server executes the directive and replaces the delimiters and the text between them with the result of executing the directive. If for some reason the server doesn't support the directive, the server ignores the directive and the browser treats the directive as a comment, which isn't displayed.

Till this point it has been stated about web page design. In the following section, the development of embedded application is described.

The application requires the *dcrtcp.lib* library, which supports TCP/IP and related protocols, and the *http.lib* library, which supports HTTP. To make embedded web server application, these libraries has to be included.

The MCS51 module has 8 channels, 12-bit Analog to digital converter, AD7870. This ADC supports programmable gain and it has a conversion time of 180µs. In this application a thermistor is used to measure the temperature and the temperature is stored as a variable, which will be attached with the web page and served to client.

```
sock_init();
 http_init();
       tcp_reserveport(80);
 while (1) {
       update_temp();
 http_handler();
 }
```

In above code segment *sock_init()* initialize the packet driver and transfer control protocol. *http_init()* initialize the http daemon. This must be called after sock_init(), and before calling http_handler() in a loop. This sets the root directory to "/" and sets the default file name to "index.html". *Tcp_reserveport* is used by servers to give special treatment to remote sockets that connect to the specified port. If a connection is made to a reserved port and there is currently not a listening socket to handle it, the system will try to hold the connection open until a listening socket is available.

This allows a server to handle multiple requests smoothly even if only a few sockets are allocated. Port number 80 is reserved for Hyper Text Transfer Protocol. Then in an infinite loop, two functions are alternatively called, in which, update_*temp()* is used to update temperature values and the library function *http_handler()* is a tick function to run the http daemon. It must be called periodically for the daemon to work.

In this work the remote control is also done, by using some user interfaces such as buttons in the web page. By clicking the button, the remote user can give commands to microcontroller, which in turn controls the relay.

HARDWARE DESIGN

The MCS51Controller is used to host web pages. This microcontroller has 5 parallel ports, six serial ports, three timer units Timer A, Timer B and Timer C.

This microcontroller has 24-bit address bus and it needs two clock inputs. One is fast clock used as processor clock and a slower clock for peripherals like Real Time Clock, Watchdog Timer, etc. This Controller reduces the EMI, though it runs at the maximum clock speed of 125 MHz. The internal blocks, Clock doubler and spectrum spreader are used to reduce EMI.

In this work Ethernet core module is used with its prototype board. This core module has Ethernet Controller, 512K Flash, 512K SRAM, an A/D converter AD7870, and optional NAND Flash.

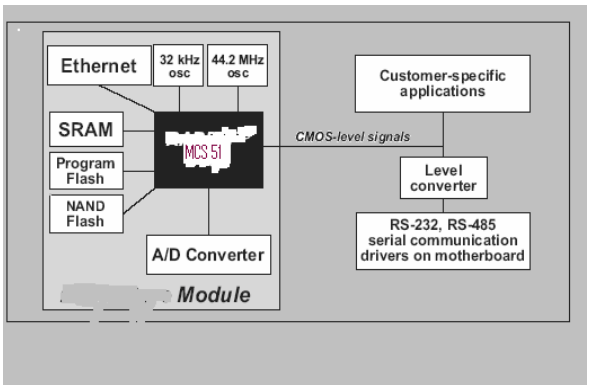

Fig. 2: Ethernet module Sub Systems

In this core module, A/D converter AD7870's 8th Channel LN7 is connected to thermister, such that the temperature variation is sensed as voltage variation.

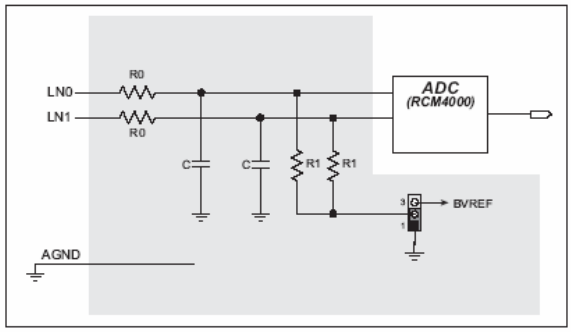

Fig. 3: Resistor Divider Network for Analog Inputs

The above diagram shows the input circuitry of the AD7870. The resistive divider acts as a 10:1 attenuator and the capacitors are used to filter out noises.

The core module has RTL8019AS Ethernet controller and RJ45 connector. The RTL8019AS is a highly integrated Ethernet Controller which offers a simple solution to implement a Plug and Play NE2000 compatible adapter with full-duplex and power down features. With the three level power down control features, the RTL8019AS is made to be an ideal choice of the network device.

The full-duplex function enables simultaneously transmission and reception on the twisted-pair link to a full-duplex Ethernet switching hub. This feature not only increases the channel bandwidth from 10 to 20 Mbps but also avoids the performance degrading problem due to the channel contention characteristics of the Ethernet CSMA/CD protocol.

TEST AND RESULTS

In this work, an embedded application is developed to serve the web page with dynamic content and with the user interfaces like buttons. The embedded application is tested in a Local Area Network. After downloading the web page and the application, by removing the programming cable the processor is put into run mode. In run mode, when the Controller is reset, it starts execution from 0 × 000000.

The board is connected to the hub via a straight through cable. It is assigned an IP address of 192.168.1.4 and a subnet Mask 255.255.255.0.

Then the processor is reset to run the application. Now in a desktop PC which is connected to the same network, a browser application is opened. In the address bar of browser, the IP address 192.168.1.4 is typed to make an http request to the Embedded Server. After getting the http request from the client, the server delivers the web page with the dynamic content.

The following figure shows the web page delivered by the Embedded Web Server.

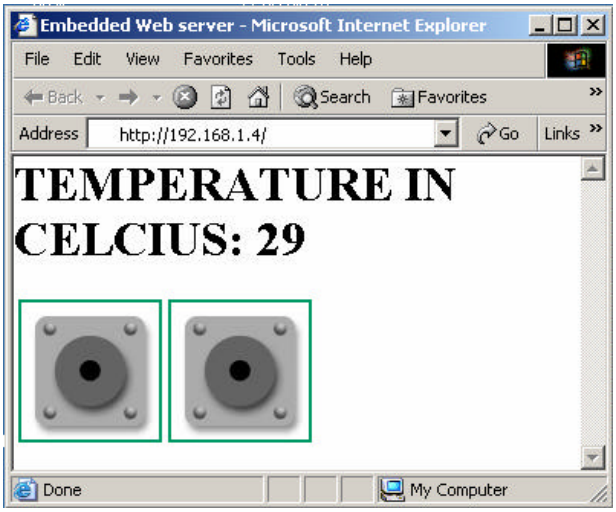

Fig. 4: Snapshot of web page served by Embedded Web Server

To connect the Ethernet kit to internet, the board has to be connected to an Internet modem with a static IP address and the web hosting service is allowed by the Internet Service Provider.

CONCLUSION & FUTURE WORK

In this paper, design of an embedded web server based on MCS51 Microcontroller is presented. The application developed in MCS51, works as a web server. The application is capable of serving the web page with dynamic content. The user in the client machine can also give command to the microcontroller by the User Interfaces in the Web page.

There are plenty of applications in which it is so useful that an Embedded System should communicate to the world through internet. For example, a vending machine which sells the cool drinks, can communicate to the distributor through internet to tell that it is going out of stock. Like this Internet communication capabilities of the Embedded Systems will start new avenues in Embedded Automation Industry.

This project can be enhanced using CAN controller for processing CAN data and using Wireless sensors to gather parameters.

Having been in the starting of 21st Century, We our-self witnesses the rate of spread of internet through out the globe. So by giving Internet Communication capabilities to your embedded system, you are providing the capability to your system to communicate with every corner of this planet.

REFERENCES

[1] Hashmi, Atif G., Mdik, Haroon W. Pervaiz, Aman and Younas., Muhammad, "Efficient and Cost Effective Approach to Control and Monitoring Area Network (CMAN)" IEEE Tr. Internet comp., 2002.

[2] Axelson., Jan, "Embedded Ethernet and Internet Complete. Designing and programming small devices for networking" Penram Publications.

[3] http:// www.Siliconlabs.com.

[4] MCS5100 user manual.

[5] Wu, Li Wei, and Hu., Jwu Sheng, "Distributed Embedded Ethernet Platform for Robots Control." IEEE International Conference on Mechatronics, Taiwan, 2005.

[6] Richard Stevens, W., "TCP/IP Illustrated", The Protocols, Volume1.